JAVA.UTIL.CALENDAR CLASS

http://www.tutorialspoint.com/java/util/java_util_calendar.htm

Introduction

The **java.util.calendar** class is an abstract class that provides methods for converting between a specific instant in time and a set of calendar fields such as YEAR, MONTH, DAY_OF_MONTH, HOUR, and so on, and for manipulating the calendar fields, such as getting the date of the next week.Following are the important points about Calendar:

- This class also provides additional fields and methods for implementing a concrete calendar system outside the package.
- Calendar defines the range of values returned by certain calendar fields.

Class declaration

Following is the declaration for **java.util.Calendar** class:

```
public abstract class Calendar
   extends Object
      implements Serializable, Cloneable, Comparable<Calendar>
```

Field

Following are the fields for **java.util.Calendar** class:

- **static int ALL_STYLES** -- This is the style specifier for getDisplayNames indicating names in all styles, such as "January" and "Jan".
- **static int AM** -- This is the value of the AM_PM field indicating the period of the day from midnight to just before noon.
- **static int AM_PM** -- This is the field number for get and set indicating whether the HOUR is before or after noon.
- **static int APRIL** -- This is the value of the MONTH field indicating the fourth month of the year in the Gregorian and Julian calendars.
- **protected boolean areFieldsSet** -- This is true if fields[] are in sync with the currently set time.
- **static int AUGUST** -- This is the value of the MONTH field indicating the eighth month of the year in the Gregorian and Julian calendars.
- **static int DATE** -- This is the field number for get and set indicating the day of the month.
- **static int DAY_OF_MONTH** -- This is the field number for get and set indicating the day of the month.
- **static int DAY_OF_WEEK** -- This is the field number for get and set indicating the day of the week.
- **static int DAY_OF_WEEK_IN_MONTH** -- This is the field number for get and set indicating the ordinal number of the day of the week within the current month.
- **static int DAY_OF_YEAR** -- This is the field number for get and set indicating the day number within the current year.

- **static int DECEMBER** -- This is the value of the MONTH field indicating the twelfth month of the year in the Gregorian and Julian calendars.
- **static int DST_OFFSET** -- This is the field number for get and set indicating the daylight savings offset in milliseconds.
- **static int ERA** -- This is the field number for get and set indicating the era, e.g., AD or BC in the Julian calendar.
- **static int FEBRUARY** -- This is the value of the MONTH field indicating the second month of the year in the Gregorian and Julian calendars.
- **static int FIELD_COUNT** -- This is the number of distinct fields recognized by get and set.
- **protected int[] fields** -- This is the calendar field values for the currently set time for this calendar.
- **static int FRIDAY** -- This is the value of the DAY_OF_WEEK field indicating Friday.
- **static int HOUR** -- This is the field number for get and set indicating the hour of the morning or afternoon.
- **static int HOUR_OF_DAY** -- This is the field number for get and set indicating the hour of the day.
- **protected boolean[] isSet** -- This is the flags which tell if a specified calendar field for the calendar is set.
- **protected boolean isTimeSet** -- This is true if then the value of time is valid.
- **static int JANUARY** -- This is the value of the MONTH field indicating the first month of the year in the Gregorian and Julian calendars.
- **static int JULY** -- This is the value of the MONTH field indicating the seventh month of the year in the Gregorian and Julian calendars.
- **static int JUNE** -- This is the value of the MONTH field indicating the sixth month of the year in the Gregorian and Julian calendars.
- **static int LONG** -- This is the style specifier for getDisplayName and getDisplayNames indicating a long name, such as "January".
- **static int MARCH** -- This is the value of the MONTH field indicating the third month of the year in the Gregorian and Julian calendars.
- **static int MAY** -- This is the value of the MONTH field indicating the fifth month of the year in the Gregorian and Julian calendars.
- **static int MILLISECOND** -- This is the field number for get and set indicating the millisecond within the second.
- **static int MINUTE** -- This is the field number for get and set indicating the minute within the hour.
- **static int MONDAY** -- This is the value of the DAY_OF_WEEK field indicating Monday.
- **static int MONTH** -- This is the field number for get and set indicating the month.
- **static int NOVEMBER** -- This is the value of the MONTH field indicating the eleventh month of the year in the Gregorian and Julian calendars.

- **static int OCTOBER** -- This is the value of the MONTH field indicating the tenth month of the year in the Gregorian and Julian calendars.
- **static int PM** -- This is the value of the AM_PM field indicating the period of the day from noon to just before midnight.
- **static int SATURDAY** -- This is the value of the DAY_OF_WEEK field indicating Saturday.
- **static int SECOND** -- This is the field number for get and set indicating the second within the minute.
- **static int SEPTEMBER** -- This is the value of the MONTH field indicating the ninth month of the year in the Gregorian and Julian calendars.
- **static int SHORT** -- This is the style specifier for getDisplayName and getDisplayNames indicating a short name, such as "Jan".
- **static int SUNDAY** -- This is the value of the DAY_OF_WEEK field indicating Sunday.
- **static int THURSDAY** -- This is the value of the DAY_OF_WEEK field indicating Thursday.
- **protected long time** -- This is the the currently set time for this calendar, expressed in milliseconds after January 1, 1970, 0:00:00 GMT.
- **static int TUESDAY** -- This is the value of the DAY_OF_WEEK field indicating Tuesday.
- **static int UNDECIMBER** -- This is the value of the MONTH field indicating the thirteenth month of the year.
- **static int WEDNESDAY** -- This is the value of the DAY_OF_WEEK field indicating Wednesday.
- **static int WEEK_OF_MONTH** -- This is the field number for get and set indicating the week number within the current month.
- **static int WEEK_OF_YEAR** -- This is the Field number for get and set indicating the week number within the current year. .
- **static int YEAR** -- This is the field number for get and set indicating the year.
- **static int ZONE_OFFSET** -- This is the field number for get and set indicating the raw offset from GMT in milliseconds.

Class constructors

S.N.	Constructor & Description
1	**protected Calendar()** This constructor constructs a Calendar with the default time zone and locale.
2	**protected Calendar(TimeZone zone, Locale aLocale)** This constructor constructs a calendar with the specified time zone and locale.

Class methods

S.N.	Method & Description

1	abstract void add(int field, int amount) This method adds or subtracts the specified amount of time to the given calendar field, based on the calendar's rules.
2	boolean after(Object when) This method returns whether this Calendar represents a time after the time represented by the specified Object.
3	boolean before(Object when) This method returns whether this Calendar represents a time before the time represented by the specified Object.
4	void clear() This method sets all the calendar field values and the time value (millisecond offset from the Epoch) of this Calendar undefined.
5	void clear(int field) This method sets the given calendar field value and the time value (millisecond offset from the Epoch) of this Calendar undefined.
6	Object clone() This method creates and returns a copy of this object.
7	int compareTo(Calendar anotherCalendar) This method compares the time values (millisecond offsets from the Epoch) represented by two Calendar objects.
8	protected void complete() This method fills in any unset fields in the calendar fields.
9	protected abstract void computeFields() This method converts the current millisecond time value time to calendar field values in fields[].
10	protected abstract void computeTime() This method converts the current calendar field values in fields[] to the millisecond time value time.
11	boolean equals(Object obj) This method compares this Calendar to the specified Object.
12	int get(int field) This method returns the value of the given calendar field.
13	int getActualMaximum(int field) This method returns the maximum value that the specified calendar field could have, given the time value of this Calendar.
14	int getActualMinimum(int field) This method returns the minimum value that the specified calendar field could have, given the time value of this Calendar.
15	static Locale[] getAvailableLocales() This method returns an array of all locales for which the getInstance methods of this class can return localized instances.
16	String getDisplayName(int field, int style, Locale locale) This method returns the string representation of the calendar field value in the given style and locale.

17	Map<String,Integer> getDisplayNames(int field, int style, Locale locale) This method returns a Map containing all names of the calendar field in the given style and locale and their corresponding field values.
18	int getFirstDayOfWeek() This method gets what the first day of the week is; e.g., SUNDAY in the U.S., MONDAY in France.
19	abstract int getGreatestMinimum(int field) This method returns the highest minimum value for the given calendar field of this Calendar instance.
20	static Calendar getInstance() This method gets a calendar using the default time zone and locale.
21	static Calendar getInstance(Locale aLocale) This method gets a calendar using the default time zone and specified locale.
22	static Calendar getInstance(TimeZone zone) This method gets a calendar using the specified time zone and default locale.
23	static Calendar getInstance(TimeZone zone, Locale aLocale) This method gets a calendar with the specified time zone and locale.
24	abstract int getLeastMaximum(int field) This method returns the lowest maximum value for the given calendar field of this Calendar instance.
25	abstract int getMaximum(int field) This method returns the maximum value for the given calendar field of this Calendar instance.
26	int getMinimalDaysInFirstWeek() This method gets what the minimal days required in the first week of the year are; e.g., if the first week is defined as one that contains the first day of the first month of a year, this method returns 1.
27	abstract int getMinimum(int field) This method returns the minimum value for the given calendar field of this Calendar instance.
28	Date getTime() This method returns a Date object representing this Calendar's time value (millisecond offset from the Epoch").
29	long getTimeInMillis() This method returns this Calendar's time value in milliseconds.
30	TimeZone getTimeZone() This method gets the time zone.
31	int hashCode() This method Returns a hash code for this calendar.
32	protected int internalGet(int field) This method returns the value of the given calendar field.
33	boolean isLenient() This method tells whether date/time interpretation is to be lenient.
34	boolean isSet(int field) This method determines if the given calendar field has a value set, including cases that the value has been set

	by internal fields calculations triggered by a get method call.
35	abstract void roll(int field, boolean up) This method adds or subtracts (up/down) a single unit of time on the given time field without changing larger fields.
36	void roll(int field, int amount) This method adds the specified (signed) amount to the specified calendar field without changing larger fields.
37	void set(int field, int value) This method sets the given calendar field to the given value.
38	void set(int year, int month, int date) This method sets the values for the calendar fields YEAR, MONTH, and DAY_OF_MONTH..
39	void set(int year, int month, int date, int hourOfDay, int minute) This method sets the values for the calendar fields YEAR, MONTH, DAY_OF_MONTH, HOUR_OF_DAY, and MINUTE.
40	void set(int year, int month, int date, int hourOfDay, int minute, int second) This method sets the values for the fields YEAR, MONTH, DAY_OF_MONTH, HOUR, MINUTE, and SECOND.
41	void setFirstDayOfWeek(int value) This method sets what the first day of the week is; e.g., SUNDAY in the U.S., MONDAY in France.
42	void setLenient(boolean lenient) This method specifies whether or not date/time interpretation is to be lenient.
43	void setMinimalDaysInFirstWeek(int value) This method sets what the minimal days required in the first week of the year are; For Example, if the first week is defined as one that contains the first day of the first month of a year, call this method with value.
44	void setTime(Date date) This method sets this Calendar's time with the given Date.
45	void setTimeInMillis(long millis) This method sets this Calendar's current time from the given long value.
46	void setTimeZone(TimeZone value) This method sets the time zone with the given time zone value.
47	String toString() This method return a string representation of this calendar.

Methods inherited

This class inherits methods from the following classes:

- java.util.Object